DOCUMENT RESUME

ED 281 519 IR 012 640

AUTHOR Fayen, Emily Gallup; And Others

TITLE The Arabization of a Full-Text Database Interface.

INSTITUTION Georgia Inst. of Tech., Atlanta.

SPONS AGENCY Agency for International Development (IDCA), Washington, DC.
PUB DATE 86
CONTRACT NEB-0016-00-2092
NOTE 5p.
PUB TYPE Reports - Descriptive (141)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Arabic; Character Recognition; Computer Software; Databases; Developed Nations; *Developing Nations; English; Foreign Countries; Machine Translation; *Man Machine Systems; *Online Searching; *Online Systems; Scientific and Technical Information; Technology Transfer; Users (Information)
IDENTIFIERS Bibliographic Retrieval Services; Character Encoding; Egypt; *Egyptian National Scientific Tech Info Network; Full Text Databases; *Nonroman Scripts; Software Design; User Cordial Interface

ABSTRACT

The 1981 design specifications for the Egyptian National Scientific and Technical Information Network (ENSTINET) stipulated that major end-user facilities of the system should be bilingual in English and Arabic. Many characteristics of the Arabic alphabet and language impact computer applications, and there exists no universally accepted character encoding scheme equivalent to the ASCII standard for Latin alphabets. In order to overcome the native language barrier in the system, a native language interface to existing software was developed. The Arabic language software functions include an Arabic editor running under the UNIX operating system, an Arabic database search facility, and electronic mail, which were implemented for peripheral devices using the CODAR/UFD Arabic character encoding scheme. The Arabic database search facility has been developed by arabizing BRS/Mate, a menu-oriented front end to the native mode of Mini-Micro BRS/Search, a full text, state-of-the-art information management software system. Five references are provided. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED281519

THE ARABIZATION OF A FULL-TEXT DATABASE INTERFACE †

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
X This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Laila H. Kamel [1]
Tarek M. Kamel [1]
Jehan Nour [1]
Theodore Hope [2]
Ra-Ed Kayyali [2]
Vladimir Slamecka [2]
Emily Gallup Fayen [3]

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY
Emily Gallup Fayen
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

ABSTRACT. - A description of the Arabic-language facilities available to users of the Egyptian National Scientific and Technical Information Network. The Arabic-language software functions include an Arabic editor running under the UNIX operating system, an Arabic database search facility, and electronic mail, implemented for peripheral devices using the CODAR/UFD Arabic character encoding scheme. The Arabic database search facility has been developed by arabizing BRS/Mate, a menu-oriented front end to the native mode of Mini-Micro BRS/Search, a full-text, state-of-art information management software system.

1. INTRODUCTION

The 1981 design specifications of the Egyptian National Scientific and Technical Information Network (ENSTINET) stipulated that major end-user facilities of the system should be bilingual (English and Arabic). This design requirement followed from the awareness that the development of computer systems capable of being used by non-programmers opened up, by late 1970's, major opportunities for extending the use of information technology to countries and nations other than those who build it. Almost simultaneously, however, came a sobering recognition of the pervasiveness of the English-language environment in which this technology had so far existed, and of the imperative need to adapt it to languages that are native to the peoples of non-English speaking countries.

The characteristics of the Arabic alphabet and language that impact computer applications are well known [1]. The alphabet consists of 28 single-case characters whose shape depends on their position in the word (context-dependency); five vowels which appear only if the meaning of a word cannot be determined from context; and various diacritic symbols that are essential only in certain kinds of text. Arabic is written from right to left; most letters in a word are joined together; and the width of a letter may vary, for aesthetic and other reasons. Numbers are also written from right to left but the most significant digit is the left one. The collation sequence of Arabic is different from that of the Latin alphabet. For data processing purposes, there exists no universally accepted character encoding scheme equivalent to the ASCII standard for Latin alphabets; rather, a number of such schemes are in use.

In principle, there are two possible approaches to overcome the native-language barrier in computer-based information systems: to develop application software in the native language, or to build a native-language interface to existing application software. So far, the former approach has been employed rarely [2]; it is costly and time-consuming, particularly in the absence of an Arabic "programmers' work bench" or similar software development tools. Eventually, however, this will be the preferred approach, as efforts such as the current internationalization of the UNIX† operating system [3]

† The work reported here was supported in part by the U.S. Agency for International Development, under contract NEB-0016-00-2092 with the Georgia Institute of Technology.

[1] ENSTINET, Academy of Scientific Research and Technology, Cairo, Egypt
[2] School of Information and Computer Science, Georgia Institute of Technology, Atlanta, Ga. 30332, USA
[3] Van Pelt Library, University of Pennsylvania, Philadelphia, Pa. 19104, USA

† UNIX is a trade mark of AT&T Bell Laboratories.

IR012640

BEST COPY AVAILABLE

encompass building environments for native-language software development.

The second approach -- adapting existing software -- may proceed in one of two ways: by modifying the software source code, or by writing program filters which act as an interface between the user's language and that of the operating software. The latter is accomplished by translating English-language messages, and selecting native-language equivalents of program tokens such as commands [4]. ENSTINET opted for the filter-based approach, partly because of the unavailability to it of the software source code, partly for reasons of contractual expediency.

This paper describes the approach taken by ENSTINET toward providing the Arabic-speaking end user with a native-language interface for building and using full text Arabic-language files and databases in a bilingual environment.

2. DEFINITION OF ARABIC SOFTWARE FUNCTIONS

Two design principles affected the selection of Arabic-language functions in ENSTINET: the intent to provide such functions only for users (system and database administrators as well as programmers are expected to use English-language software facilities); and the intent that ENSTINET be used directly by problem solvers ("knowledge workers"), rather than by database professionals alone. In such an end-user community, some will be multilingual, some bilingual, and others monolingual. ENSTINET's concern is with users whose native language is Arabic, and who may or may not be proficient in English. The ENSTINET "bilingual environment" is thus designed to support users who communicate, in Arabic, with predominantly Arabic bodies of information; and those who communicate, in English, with predominantly English bodies of information.

Accordingly, the Arabic-language capabilities selected for ENSTINET end-users were: Arabic text editing; searching Arabic databases; and electronic mail. (As of summer 1986, electronic mail remains to be implemented.)

3. ARABIC SOFTWARE DEVELOPMENT ENVIRONMENT

The implementation of these arabized software capabilities took place in a development environment consisting of an AT&T 3B2/400 computer running UNIX System V, a Multi-Media Video Arabdata-50 terminal (bilingual Wyse-50), and an Arabic text editor.

The Arabdata-50 (as well as the ENSTINET printers) use the 7-bit CODAR/UFD character encoding scheme [5], one of the more widely accepted standards that maps Arabic characters to the alphabetic ASCII characters, and digits and punctuation to their corresponding ASCII codes. Because of this mapping, bilingual text can only be produced by embedding control characters or escape sequences in the text. These "language switchers" map to commands which, when sent to a terminal or printer, instruct it to switch to the corresponding language set.

(Parenthetically, embedding language switching commands in text is not the optimal form of storing information -- and one of the reasons why ENSTINET databases are monolingual (English or Arabic), at least for the time being. Preferred ways for designating the language of text is to use the most significant bit of every character, or employ two-byte character codes for multilingual text. Such schemes allow to determine the language of a block of text, or even of a single character, to be determined regardless of context. These approaches are to be supported in UNIX System V Release 3.1.)

The Arabdata-50 operates in either Arabic or English mode, set via escape sequences. In English mode, it behaves like a standard ASCII terminal; in Arabic mode, Arabic characters are displayed from right to left. In addition, the terminal supports the ability to enter another language mode temporarily; that is, if it is in Arabic mode, the "enter temporarily English mode" control character instructs the terminal to display all subsequently received characters on that line in English. While in this mode, the English characters are "pushed in" from right to left, to preserve the overall Arabic structure of the line. The "exit temporary English mode" control character reverts the terminal back to the Arabic mode. Analogous control characters exist for entering and leaving "temporary Arabic mode" from English mode. These temporary language escape characters allow bilingual text to be mixed and displayed on the screen.

In the absence of a commercially available Arabic editor running under the UNIX operating system, the Georgia Institute of Technology developed two versions of such an Arabic editor. The first one, *ared*, is a modified version and Arabic equivalent of the UNIX line editor, *ed*. It provides most of the functions available in *ed*, and accepts commands in Arabic. Its main use has been for entering and modifying large blocks of Arabic text. The second one, *bed*, is similar to *ed* but it also accepts Arabic substrings (delimited by the appropriate control characters) embedded within lines of English text. Independently of their use in the Arabic software development, these editors permit Arabic users to build and modify Arabic and Arabic-English text files.

4. ARABIC DATABASE SEARCH INTERFACE

ENSTINET database software is BRS/Search‡ Mini-Micro Version, a state-of-art, full-text information management product running under the UNIX operating system.

BRS/Search provides several user interfaces; the primary is the *Native* user interface, comprising the processes which the other interfaces execute. All interfaces provide for database que modification, and printing; the difference among them lies in the degree of user-friendliness. The BRS/Mate, chosen as the target for arabization, is designed for the relatively inexperienced user; it is basically a menu-based front end to the *Native* search interface, consisting of a collection of functions (menus) executed by an interpreter program, *brsmns*. Each menu corresponds to one UNIX file written in a high-level language, MNS. The latter provides constructs such a control structures, global variables, conditional and unconditional branches, function calls (calls to other menus), and sophisticated string manipulation in the form of parsing and pattern matching. An MNS program interacts with the user via terminal input and output statements; on the other end, it communicates with the *Native* mode search program.

Version 3 of BRS/Search, released in 1985, supports the use of multiple character sets, and allows to define the collating sequence of each character set. These and other enhancements (such as support for different terminal devices) of the BRS/Search loader have made it possible to approach the task of arabization without the need to modify the underlying software. Many of the component parts of BRS/Search, such as output messages, commands, and logical search operators, are not hard-coded but are defined in files and associated with a number; converting the English version of BRS/Mate to Arabic consisted largely of creating parallel, translated files for use by the Arabic MNS menus. In a number of instances, however, different solutions had to be developed, as follows:

- BRS/Mate maps all tokens entered by the user, including search statements, into upper case. Since Arabic is a single-case alphabet, the Arabic user's input would have been mapped by BRS/Mate into other, completely different Arabic characters. Accordingly, certain parsing statements had to be removed from many MNS menus, and extra logic added to ensure correct mappings when they were needed.
- The system date and time, which are provided to MNS in the form of an English string containing the day of the week, month, time, and year, were converted to their Arabic equivalent by a set of CASE statements.
- Variables in MNS menus can be set either by direct user input, assignment statements, or return status of internal commands. In many instances, the value which a variable can take on determines the flow of the program. Since the input from an Arabic user is a character that does not necessarily correspond to expected variable values, many segments of MNS code had to be modified by using CASE statements to selectively map Arabic input options to their corresponding English ones.
- Escape or control character sequences have been added to many of the translated terminal output (PRINT) statements, instructing the terminal to change to the appropriate language mode. This was necessary because even though the interface is almost entirely in Arabic, certain inaccessible, low-level functions of BRS/Search output English messages.

‡ BRS/Search and BRS/Mate and trademarks of BRS Technologies.

5. CONCLUSION

The effort described in this paper has been aimed at providing a native-language database search capability particularly suited for ENSTINET end-users. The arabized BRS/Mate interface is a user-friendly, state-of-art product that fully satisfies this objective. The approach taken has been highly expedient, the effort was completed within the resources allocated (about four man-months over a three-month period), and it was accomplished without resort to the source code. The techniques used, however, should not necessarily be viewed as being generally applicable to native-language software development. The announced internationalization of the operating system UNIX should allow, in the near future, more generic approaches to the development of native-language software.

Under review by ENSTINET is also the issue of a single-language search interface to bilingual and multilingual databases. Although ENSTINET at present has no effort underway to build such databases, partially because of the cumbersome manner of designating the text languages in the database (as discussed above), they could eventually be generated relatively simply by merging two or more monolingual databases. With the present Arabic BRS/Mate interface, a bilingual user would retrieve only the Arabic-language records; in other words, two search passes (Arabic and English) would be necessary.

An alternate approach to searching multilingual databases is to use multilingual dictionaries, with software that automatically maps a string in one language into its equivalents in other languages. ENSTINET, which uses free-text search software, could in principle employ an English-Arabic dictionary, automatically create a bilingual inverted index to the bilingual database, and use a single-language interface to search both languages. Since ENSTINET's software uses free-text indexing and searching, ENSTINET does not contemplate undertaking the very expensive effort of building and maintaining multilingual thesauri, partly because the effort is very expensive, partly because thesauri are impractial as tools for end-user database searching.

6. REFERENCES

[1] See, for example, Amman, M. A., "Use of Arabic in Computerized Information Exchange." *J. Am. Soc. Information Science*, July 1984, pp. 204-210.

[2] Graham, D., "Saudi-Built Computer Programs in Arabic." *Saudi Business*, 7:22-25 (1983).

[3] AT&T. *Native Language Supplement for UNIX System V.* AT&T, PM/AC-3005 (1986).

[4] Santaella, E.M. and Slamecka, V., "Toward Native Language Software for Information Management." *Information Processing & Management*, 20(4):527-534 (1984).

[5] *Arab Journal of Science*, 1(1):99 (1982).